Mission Pinpossible Writeup - Hardware Challenge

By run3 (respect in https://www.hackthebox.eu/profile/223654)

You are given 2 files, which are op_pinpossible and Security Keypad

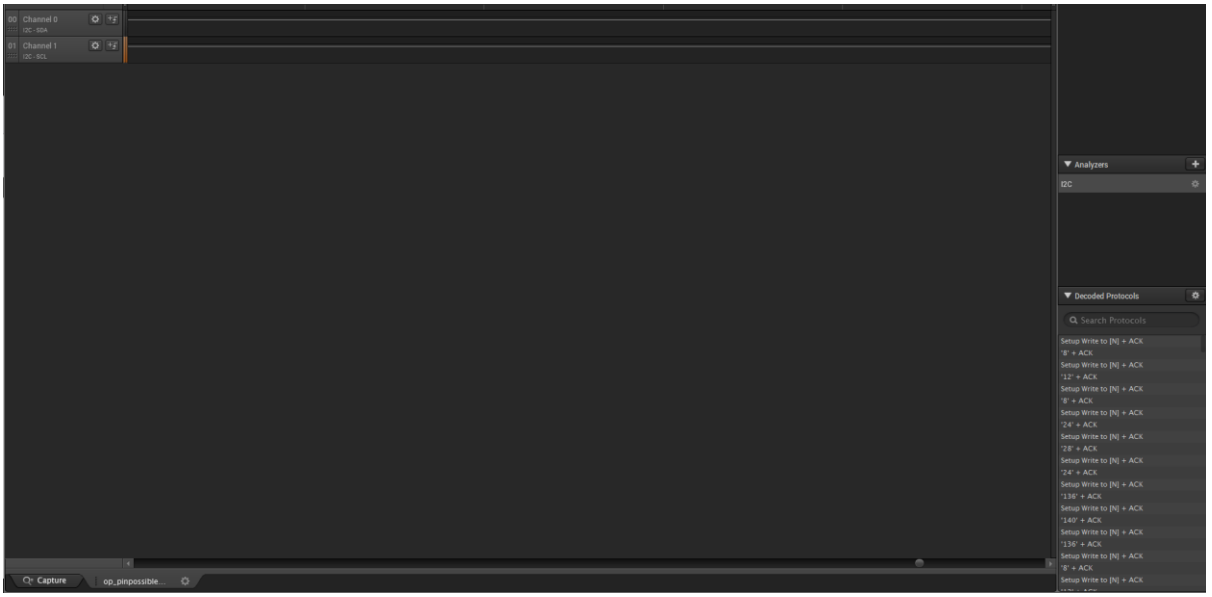
Open up Security Keypad.jpg

Security Keypad

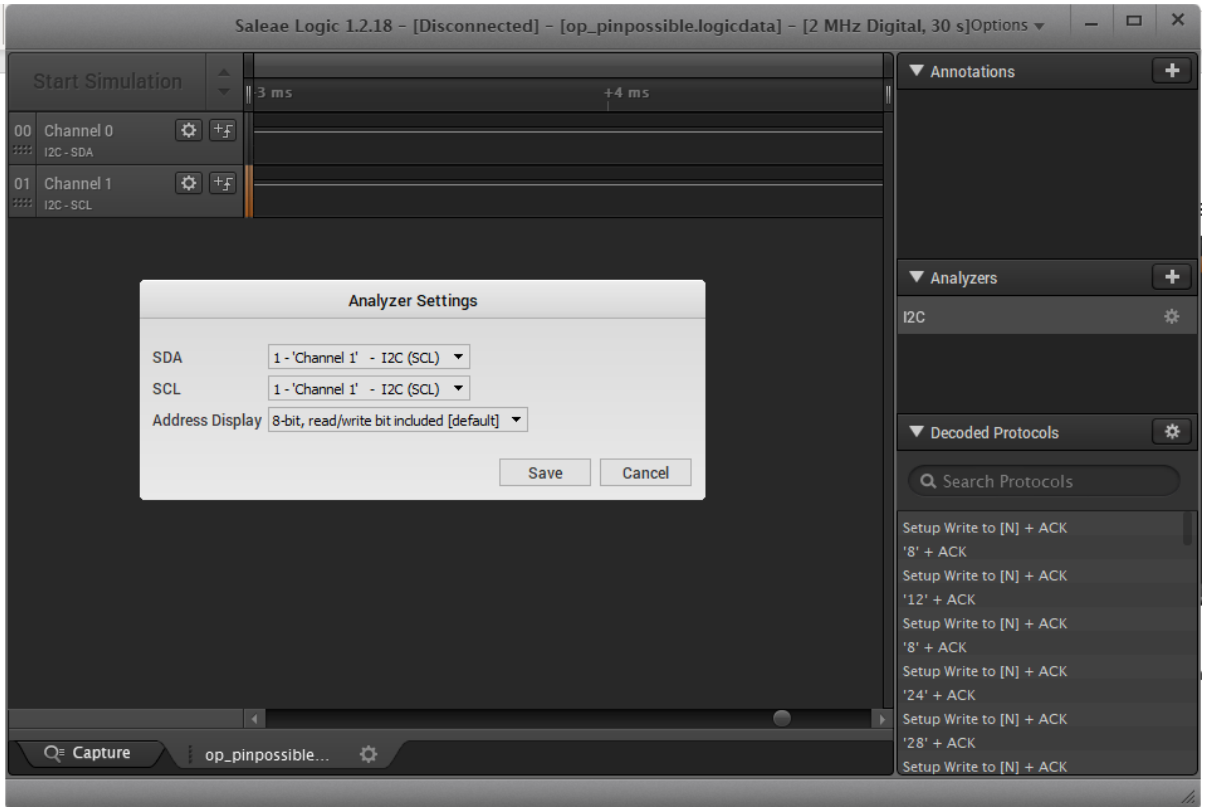
LCD Display Internal Photo

We are Presented by this zooming in to the chip we can see that it is PCF8574T. Searching what is the chip all about I stumbled across a youtube video https://www.youtube.com/watch?v=NxbFrGxGgwM which says that this chip is used for i2c which is a display lcd for a Arduino.

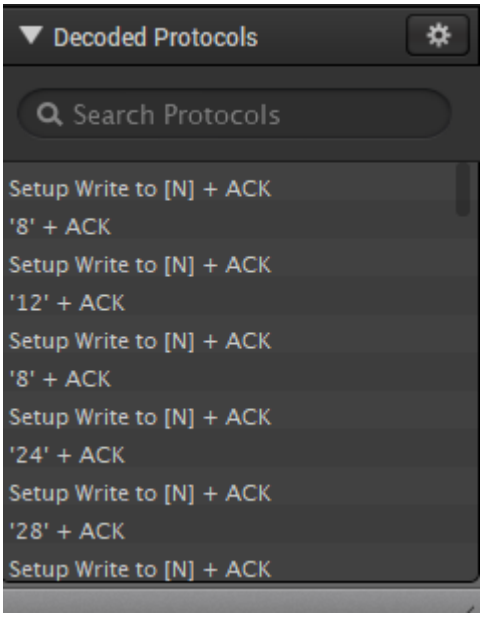
Now Using Saleae Logic to open op_pinpossible.logicdata



As we can see there are 2 channels, channel 0(i2c-SDA) and 1(i2c-SCL) , then I saw this Analyzer tab in the right side we are going to add i2c



We now can see that we have data in the tab of Decoded protocols

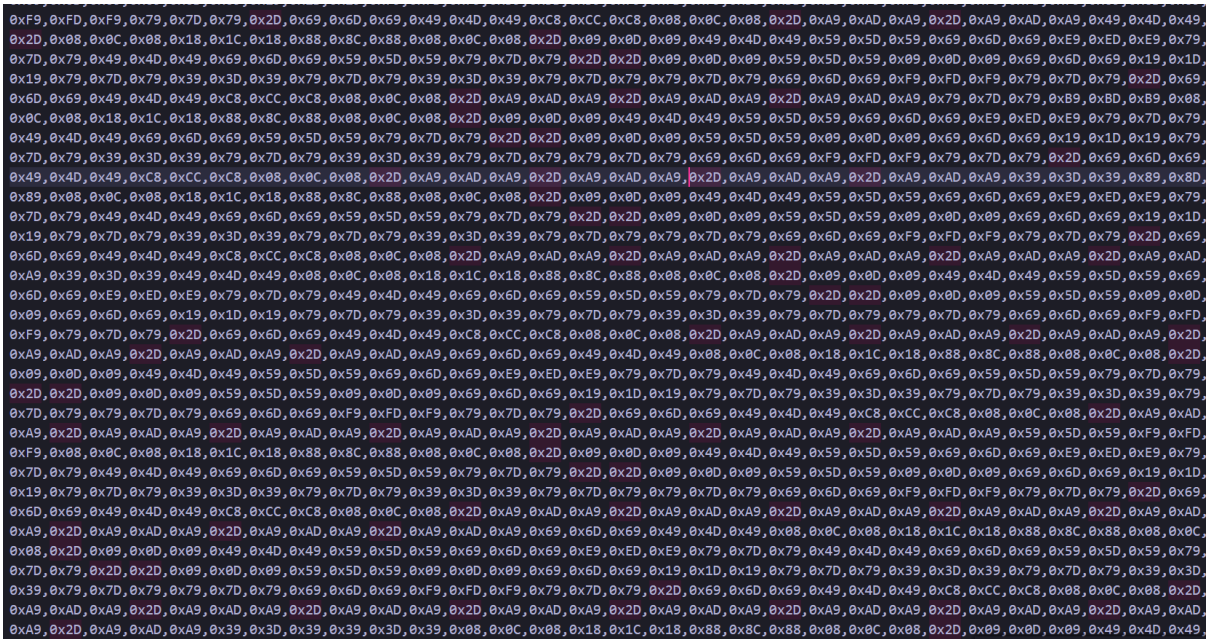


Changing the Display Radix to Ascii & Hex and Export as .csv

```
Time [s],Packet ID,Address,Data,Read/Write,ACK/NAK
0.448499000000000,0,N (0x4E),'8' (0x08),Write,ACK
0.448728500000000,1,N (0x4E),'12' (0x0C),Write,ACK
0.448958500000000,2,N (0x4E),'8' (0x08),Write,ACK
0.449248000000000,3,N (0x4E),'24' (0x18),Write,ACK
0.449478000000000,4,N (0x4E),'28' (0x1C),Write,ACK
0.449707500000000,5,N (0x4E),'24' (0x18),Write,ACK
0.452084500000000,6,N (0x4E),'136' (0x88),Write,ACK
0.452314500000000,7,N (0x4E),'140' (0x8C),Write,ACK
0.452544000000000,8,N (0x4E),'136' (0x88),Write,ACK
0.452823500000000,9,N (0x4E),'8' (0x08),Write,ACK
0.453053500000000,10,N (0x4E),'12' (0x0C),Write,ACK
0.453293000000000,11,N (0x4E),'8' (0x08),Write,ACK
0.453587500000000,12,N (0x4E),) (0x29),Write,ACK
0.453817500000000,13,N (0x4E),- (0x2D),Write,ACK
0.454047000000000,14,N (0x4E),) (0x29),Write,ACK
0.454337000000000,15,N (0x4E),\t (0x09),Write,ACK
0.454566500000000,16,N (0x4E),\r (0x0D),Write,ACK
0.454796000000000,17,N (0x4E),\t (0x09),Write,ACK
0.455081000000000,18,N (0x4E),I (0x49),Write,ACK
0.455316000000000,19,N (0x4E),M (0x4D),Write,ACK
0.455545500000000,20,N (0x4E),I (0x49),Write,ACK
0.455830500000000,21,N (0x4E),Y (0x59),Write,ACK
0.456060000000000,22,N (0x4E),] (0x5D),Write,ACK
0.456289500000000,23,N (0x4E),Y (0x59),Write,ACK
0.456574500000000,24,N (0x4E),i (0x69),Write,ACK
0.456804000000000,25,N (0x4E),m (0x6D),Write,ACK
0.457034000000000,26,N (0x4E),i (0x69),Write,ACK
0.457313500000000,27,N (0x4E),'233' (0xE9),Write,ACK
0.457543000000000,28,N (0x4E),'237' (0xED),Write,ACK
0.457773000000000,29,N (0x4E),'233' (0xE9),Write,ACK
0.458057500000000,30,N (0x4E),y (0x79),Write,ACK
```

We are Presented this data after we export as .csv,

While Reading the file given to us we can see that all the data is being written so it always sends the data into the i2c, so I just need to leave the data column and remove everything



using Excel and Words to remove the Spaces and unwanted Characters.

We are presented like this.

now we can work with this data in python and start to decode it. the easiest way to understand what is going on for me is to read the code of an existing library made to communicate with this kind of setup. So I picked this lib as my guide

https://github.com/mathertel/LiquidCrystal_PCF8574/blob/master/src/LiquidCrystal_PCF8574.cpp

as we can see, we send our data in `_send`, where it gets split up into two half bytes ("nibbles") that get sent as separate messages with two `_sendNibble` calls and then finally the data actually gets transferred in `_write2Wire` here we can see that our data gets shiften to the 4 most significant bits of the message and the flags `PCF_RS` and `PCF_EN` get set if we send data and the signal is set to enabled. since we've got this data as an easy to read list in python.

So I made a Script

```
packets = [0x08,0x0C,0x08,0x18,0x1C,0x18,0x88,0x8C…..]
PCF_RS = 0x01
PCF_EN = 0x04


interesting_packets = []
for packet in packets:
    if (packet&PCF_RS) and (packet&PCF_EN):
        interesting_packets.append(packet)
paired_packets = [interesting_packets[i:i + 2] for i in range(0, len(interes
ting_packets), 2)]

for upper_nibble,lower_nibble in paired_packets:
    val = (upper_nibble&0xF0)|(lower_nibble>>4)
    print(chr(val),end="")
```

After Running this script we are presented

Now we can see the text are readable already and there is the flag just combine them

HTB{84d_d3519n_c4n_134d_70_134k5!d@}